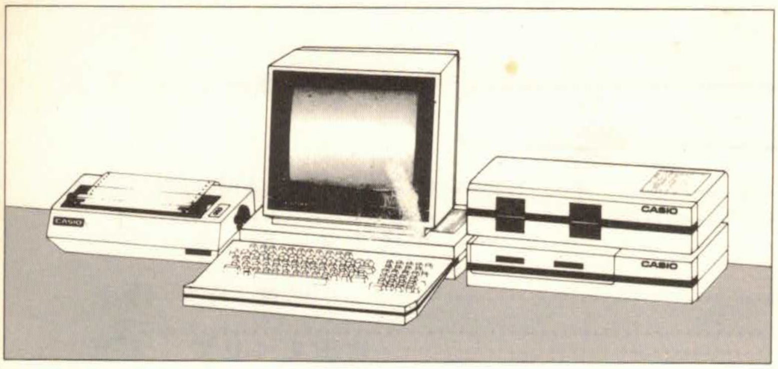


CASIO

PERSONAL COMPUTER

FP-1000シリーズ

パーソナルコンピュータ
FP-1000/1100
ハードウェア解説書

カシオ計算機株式会社

目　次

§1　本書の性格……1

§2　ハードウェア概要……2

§3　アドレスマップ……4
3-1　メインCPUブロックの構成……4
3-2　メモリーマップ……6
3-3　I／Oマップ……8
3-3-1 インタラプトレジスタ……8
3-3-2 CPU間転送ポート……9
3-3-3 ページ切換レジスタ……9
3-3-4 ページ切換レジスタ(汎用拡張ボックス内)……10
3-3-5 デバイスコード……10
3-3-6 FDD……11
3-3-7 RAMパック……12
3-3-8 ROMパック……12
3-3-9 RS-232Cパック……13
3-3-10 スロットの指定方法……14

§4　インタフェイス……16
4-1　カラーモニタ・インタフェイス……16
4-2　グリーンモニタ・インタフェイス……18
4-3　カセット・テープレコーダ・インタフェイス……19
4-4　プリンタ・インタフェイス……20
4-5　汎用拡張スロット・インタフェイス……22

§5　CALL文，USR関数の利用法……25
5-1　BASIC内のデータ表現……25
5-2　CALL文……27
5-3　USR関数……30

§6　C_{82}-BASICのファイル管理……32
6-1　FDDのファイル管理……32
6-2　RAMパック，ROMパックのファイル管理……35

§7　パワーオンブート……37

§8　C_{82}-BASICエントリーポイント……38
8-1　キーボード入力……38
8-2　CRT表示……39
8-3　プリンタ出力……43
8-4　CMT(カセットテープレコーダ)……44
8-5　RS-232C……46
8-6　FDD……48
8-7　演算サブルーチン……49
8-7-1 型変換……49
8-7-2 四則演算……50
8-7-3 関数……51
8-7-4 文字数値変換……51

§1．本書の性格

FP-1000／FP-1100を徹底的にご活用いただくために本書を編みました。

本書はFP-1000／1100を通常に使用される限りでは読む必要はありません。特にマシン語を使用し、マイクロコンピュータシステムとしてお使いになる際に十分役立つ情報が記載されています。従って、マイクロコンピュータのハードウェア・ソフトウェアに関して深い知識と経験がある方を想定して記述してあります。

本書を十分ご理解いただく為には、以下の知識が前提となります。

(1)マイクロコンピュータZ80のハードウェアの知識

(2)マイクロコンピュータZ80のマシン語の知識

(3)C_{82}-BASIC

また、FDDとRS-232Cに関しては以下の知識が必要です。

(1)FDC(μPD765AC)の知識

(2)USART(μPD8251AC)の知識

急がば廻れといいます。これらを十分ご修得の上、本書にお取組下さい。

※本書の内容につきましては解説等致しかねますので、あらかじめご了承ください。
尚、編集には万全を期しておりますが、万一誤記等ありましたら、弊社までご連絡いただきますようお願い申し上げます。

§2. ハードウェア概要

8ビットCPU 2個を使用した、マルチプロセッサ構成になっています。

《メインCPUブロック》

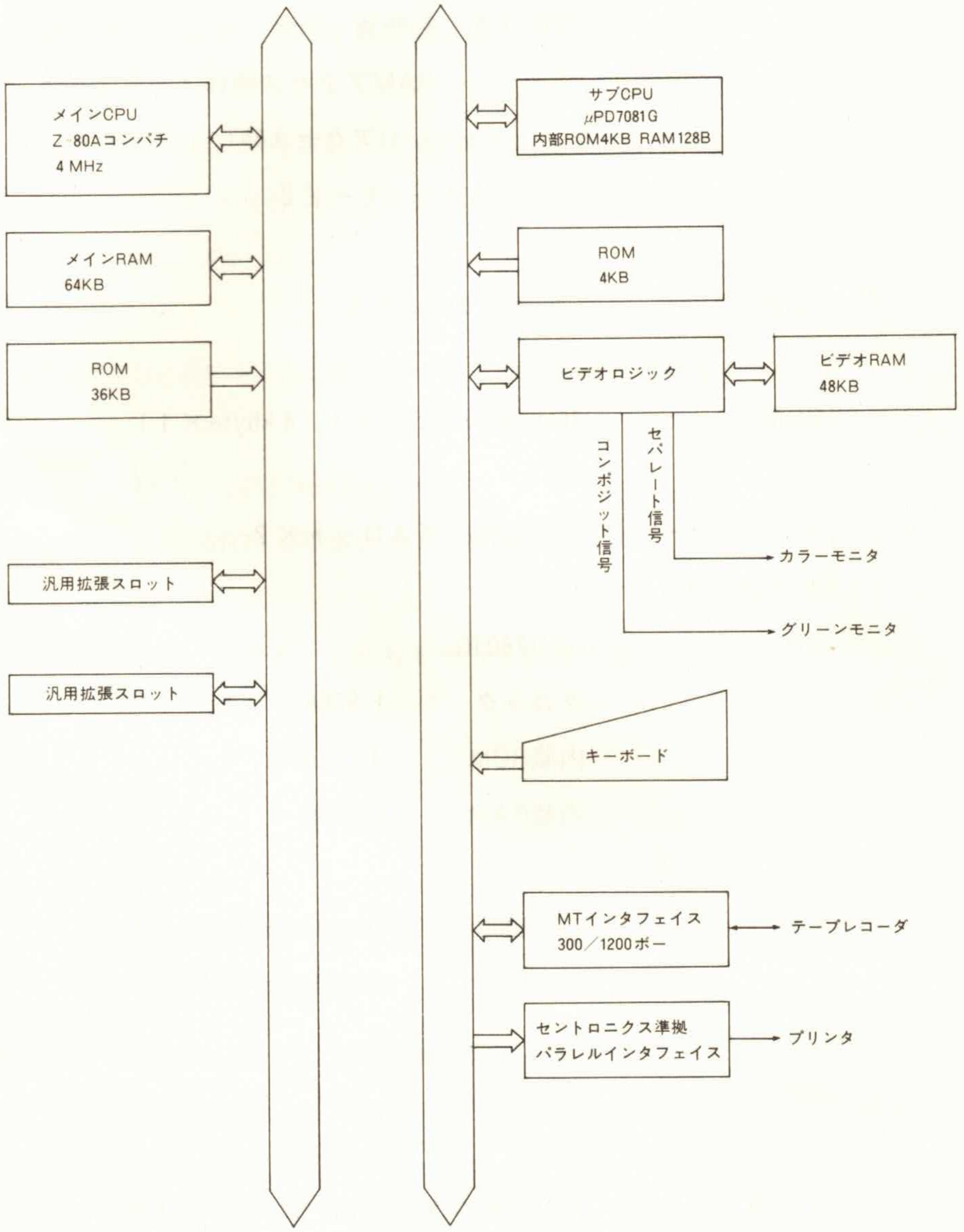
(CPU)	Z80Aコンパチブル	
	クロック：3.9936 MHz	
		RAMアクセス時はノーウェイト
		I/Oアクセス時は1ウェイト
	インタラプト：モード2	
(RAM)	64kbyte(64kbit×8)	
(ROM)	36kbyte(32kbyte×1, 4kbyte×1)	
(インタフェース)	汎用拡張バススロット×2	

《サブCPUブロック》

(CPU)	μPD7801G	
	クロック	1.9968 MHz
	内蔵ROM	4kbyte
	内蔵RAM	128byte
(RAM)	FP-1000	16kbyte(ビデオRAM)
	FP-1100	48kbyte(ビデオRAM)
(ROM)	4kbyte	
(インタフェース)	カラーモニタ (RGBセパレート信号)	

グリーンモニタ(コンポジット信号)	
プリンタ	(セントロニクスインタフェイス準拠)
カセット	(オーディオ・インタフェイス)
(キーボード)	JIS標準配列準拠(キーボード分離型)
	ステップスカルプチュア・キーボード

〈FP-1000/1100システム構成図〉

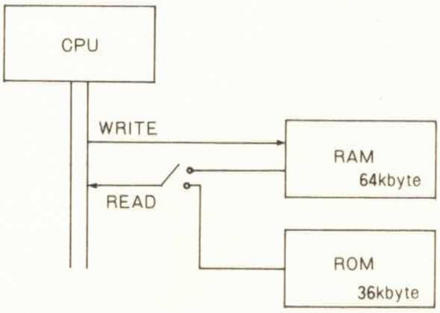


§3. アドレス・マップ

FP-1000／1100はメインCPUとサブCPUからなるマルチプロセッサ構成となっています。サブCPUはお客様が直接制御することはできません。したがって、この章では、メインCPUのアドレスマップを紹介いたします。

3-1 メインCPUブロックの構成

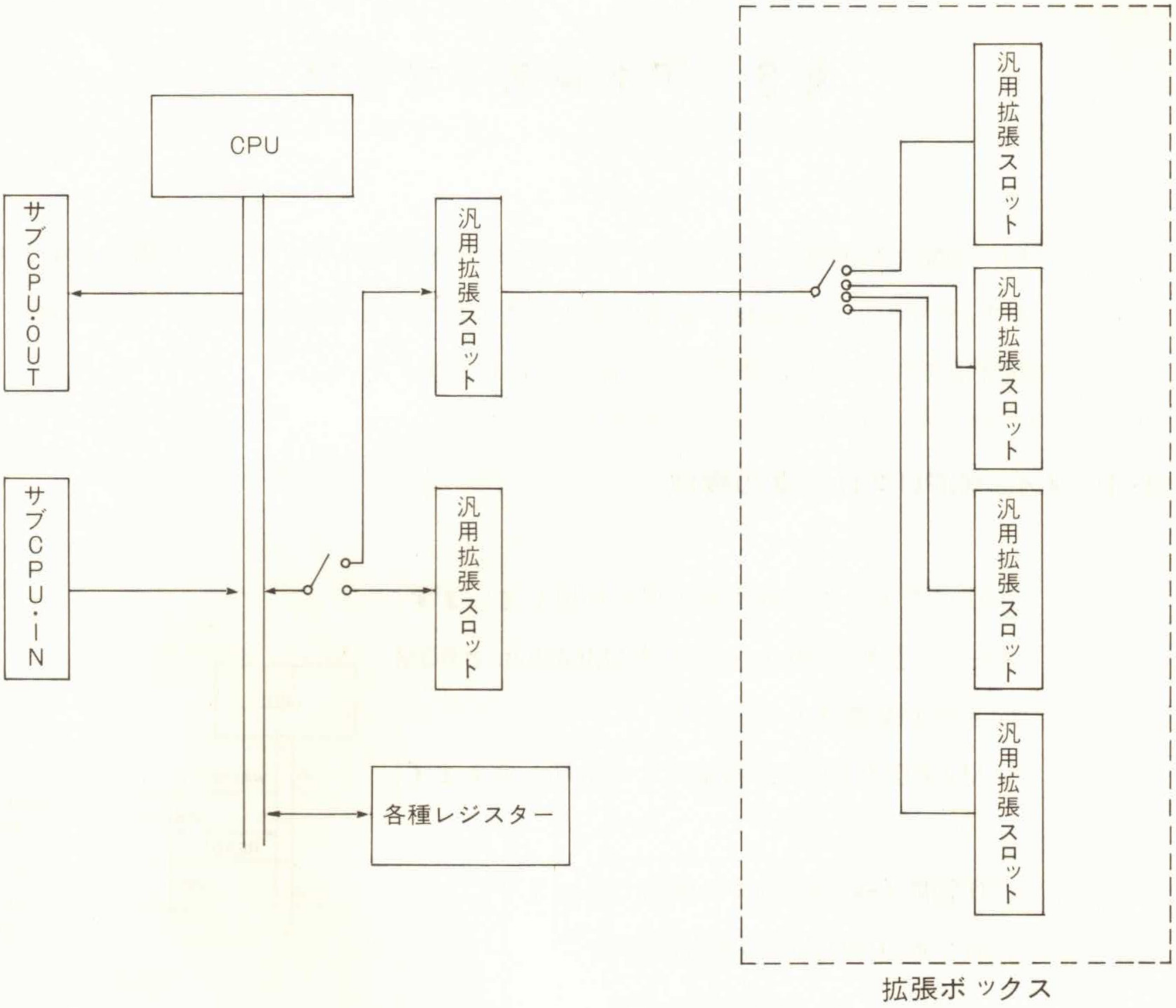
CPU のアドレス空間を最大限に利用しています。
メモリー空間の64kbyteには、RAM64kbyteとROM 36kbyteが配置されています。
RAMとROMはページ指定によって切り換えます。



I／O空間は64kbyte全体を使用しています。
このため、CPUのI／O命令のうち

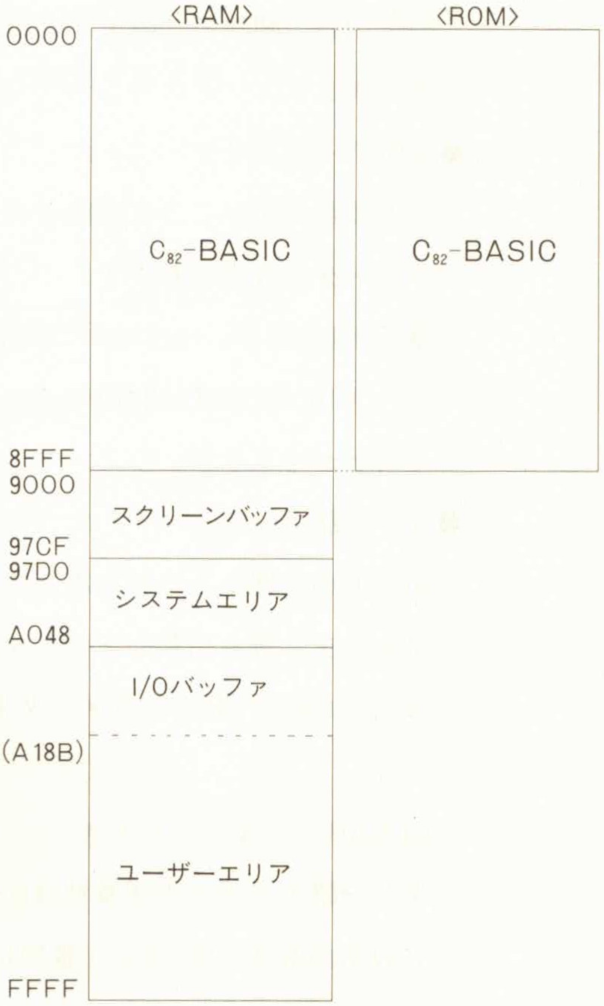
IN A, n と OUT n, A

の使用は禁じられます。(アドレスの上位にAcc.Aの内容が出力されてしまうからです。)
I／O空間の64kbyteには、汎用拡張スロットが2スロット(汎用拡張ボックスFP-1060IOを使用した場合には、最大8スロット)と、ページ切換レジスタ，サブCPUとのIOポート，インタラプトコントロールレジスタが配置されています。
汎用拡張スロットは各々0000H～0FEFFHまでの63.75kbyteのアドレス空間を持ち、ページ指定により切換えます。


CPU
サブCPU・OUT
サブCPU・IN
汎用拡張スロット
汎用拡張スロット
各種レジスター
汎用拡張スロット
汎用拡張スロット
汎用拡張スロット
汎用拡張スロット
拡張ボックス

3-2 メモリーマップ

電源ONでROMが指定され、C82-BASICをRAMへコピーします。それ以降は全てRAM上で動作します。



●スクリーンバッファ

CRT表示の文字内容をそのまま記憶しています。スクリーンエディタは実際には、このスクリーンバッファに対して働きます。

●システムエリア

C82-BASICが使用しているワークエリアです。

●I/Oバッファ

入出力用のバッファエリアです。MOUNT文での指定により、大きさが変化します。

●ユーザーエリア

プログラム,データなどユーザーが使用できる領域です。先頭からプログラムが格納されます。

先頭アドレスはI/Oバッファの大きさにより変化します。9DE8H,9DE9H番地に先頭アドレスが格納されています。

ユーザーエリアは先頭から、プログラム・データエリア,文字データエリア,マシン語エリアの順に分割されます。

●プログラム・データエリア

BASICで書かれたプログラム，数値変数等が入ります。FRE(数式)関数でこのエリアの未使用領域を求めることができます。

●文字データエリア

文字変数の値がここに格納されます。CLEAR文の第１パラメータで設定されます。

電源ON直後は、1023byteに設定されています。

また、FRE(文字式)関数でこのエリアの未使用領域を求めることができます。

●マシン語エリア

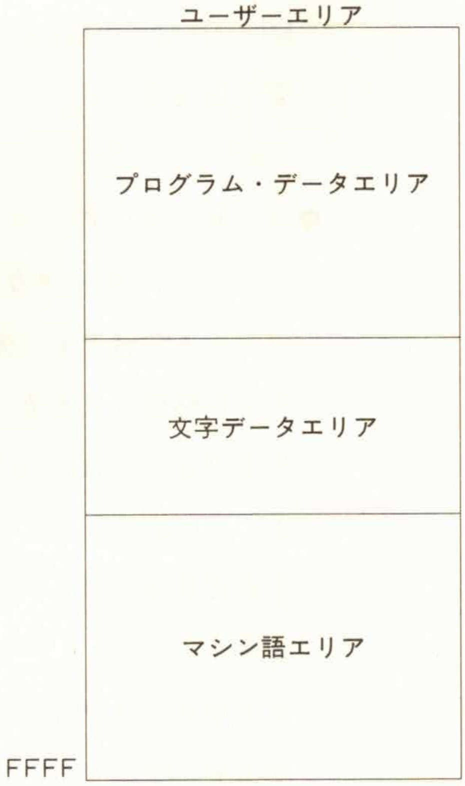
BASICの管理から外れたエリアです。ここにマシン語サブルーチン等を格納します。マシン語エリアの先頭番地はCLEAR文の第２パラメータ＋１番地です。

CLEAR　A, Bとしたとき

マシン語エリアの先頭番地はB＋１番地，文字データエリアの先頭番地はB－A＋１番地になります。

I/ Oバッファの数とユーザーエリアの先頭

n	アドレス
0	A18B
1	A2CD
2	A40F
3	A551
4	A693
5	A7D5
6	A917
7	AA59
8	AB9B
9	ACDD
10	AE1F
11	AF61
12	B0A3
13	B1E5
14	B327
15	B469



3-3 I／Oマップ

64kbyteのI／O空間のうち、0000H～0FEFFHのアドレスは、汎用拡張スロットに割当てられています。これはページ切換えで使用されます。0FF00H以降に各 I／O ポートが割付けられています。

3-3-1 インタラプトレジスタ（0FF80H～0FF9FH）

FF80 ～ FF9F	D7	D6	D5	D4	D3	D2	D1	D0
(WRITE)	SUB INT	×	×	INTS	INTD	INTC	INTB	INTA

D4～D0:

0	インタラプト禁止
1	インタラプト許可

D7:

0	サブCPU割込解除
1	サブCPU割込発生

インタラプトベクトルは以下の様に使用しています。

INTS：(F0)（コード）サブCPUからのインタラプト

INTA：(F2) FDDコントローラで使用しています。

INTB：(F4) FDDコントローラで使用しています。

INTC：(F6) RS232Cコントローラで使用しています。

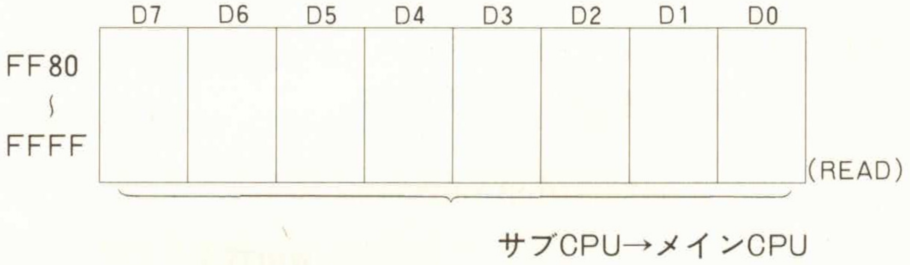
INTD：(F8)

インタラプトは、モード2の割込を使用しています。Iレジスタは00Hです。優先順位は、INTS＞INTA＞INTB＞INTC＞INTDとなっています。

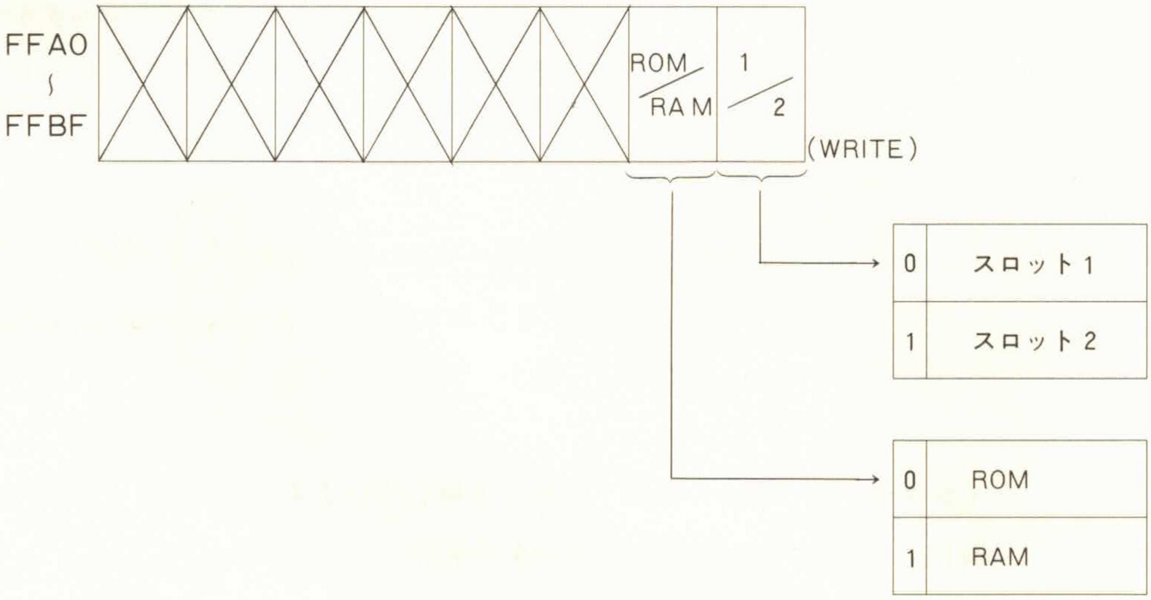
また、D7を1とすることで、サブCPUへ割込をかけることができます。

通常は、このレジスタを使用しません。

3-3-2　CPU間転送ポート

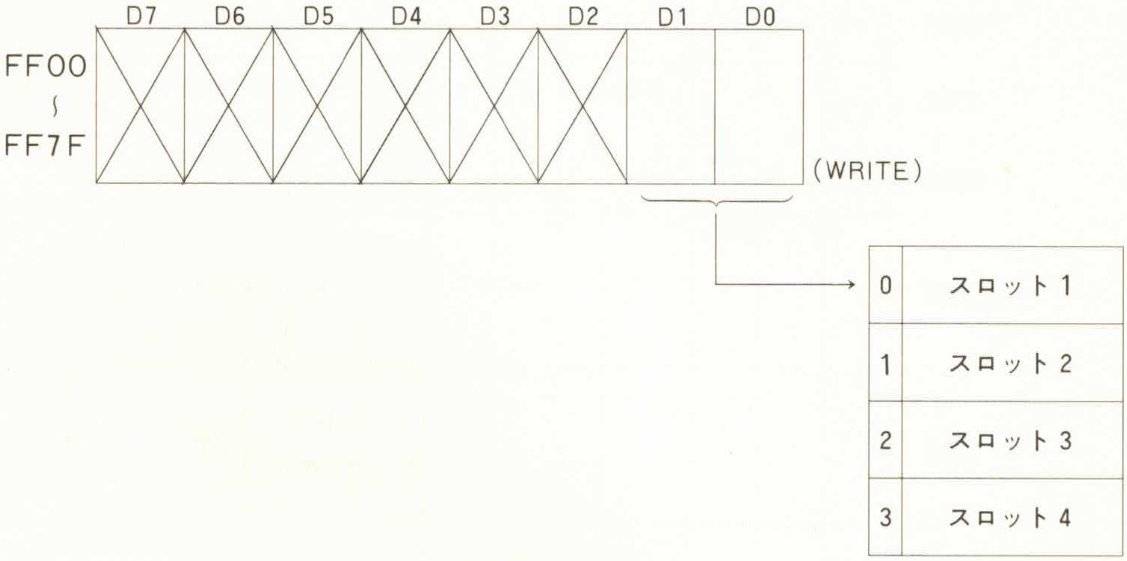


3-3-3　ページ切換レジスタ



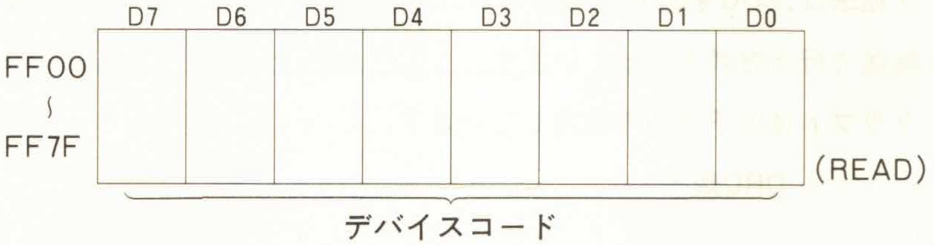
3-3-4 ページ切換レジスタ(汎用拡張ボックスFP-1060IO)

汎用拡張スロットのI／O空間をさらに4頁に切り換えます。



3-3-5 デバイスコード

現在指定しているスロットからデバイスコードを読取ります。



デバイスコードは、スロットに装着されるパックの種類を示すコードで、指定したパックに$\overline{CS}$信号を出力するとパックから出力されます。

デバイスコードは以下の通り決められています。

●FP-1020FD	ミニフロッピーディスク	04H
●FP-1030	RAMパック	01H
●FP-1031	ROMパック	00H
●FP-1035RS	RS-232Cパック	02H
●実装されていないとき		0FFH

3-3-6 FDD(FP-1020FD)

Address	D7	D6	D5	D4	D3	D2	D1	D0	R/W	内容
0000 ~ 0001									W	モーターオン。一度アクセスすると、1分間モーターが廻り続けます。
0002 ~ 0003									W	TCパルス。FDCのTC端子にパルスを出力します。
0004									R	FDCステータスレジスタ
0005									R/W	FDCデータレジスタ
0006									R/W	FDCデータレジスタ、DACK信号を自動的に付加します。

FDCはμPD765ACを使用しています。

データ転送は、FDCをDMAで動作させ、CPUは、INTA信号をチェックしながらプログラム転送で行う方式をとっています。

インタラプトは以下の通り使用しています。

INTA············DRQ端子

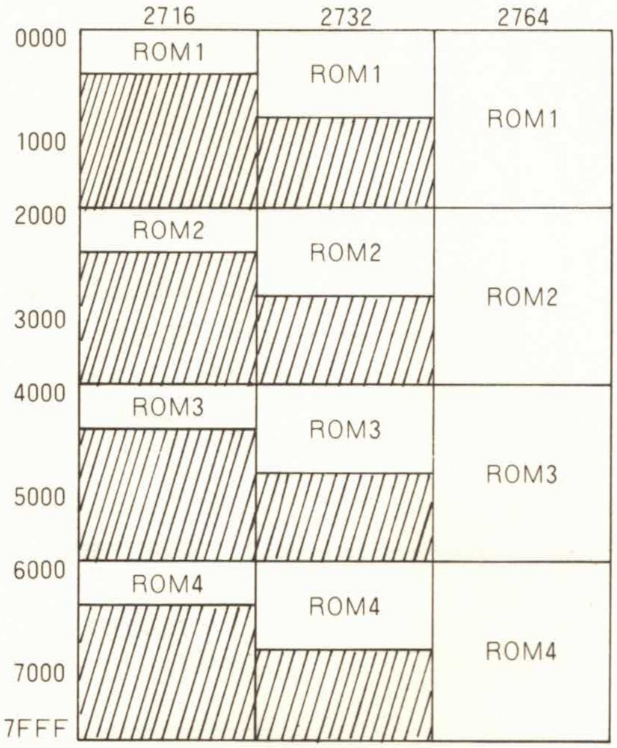
INTB············INT端子

3-3-7　RAMパック(FP-1030)

0000H～3FFFHにRAMが有ります。

3-3-8　ROMパック(FP-1031)

0000H～7FFFHにROM1～ROM4が配置されます。



3-3-9 RS-232Cパック(FP-1035RS)

	D7	D6	D5	D4	D3	D2	D1	D0		
0000									R/W	USARTのコマンドレジスタ
0001									R/W	USARTのステータスレジスタ
0002	$\overline{\text{DIPSW}}$ No.4	X	X	$\overline{\text{CTS}}$	TxRDY	X	X	$\overline{\text{CD}}$	R	

D0 ($\overline{\text{CD}}$):

0	CD有り
1	CD無し

D3 (TxRDY):

0	TxRDY無し
1	TxRDY有り

D4 ($\overline{\text{CTS}}$):

0	CTS有り
1	CTS無し

D7 ($\overline{\text{DIPSW}}$ No.4):

0	DIPSW4ON
1	DIPSW4OFF

USARTにはμPD8251ACを使用しています。

インタラプトは以下の通り使用しています。

INTC------------RXRDY端子

3-3-10 スロットの指定方法

(1)汎用拡張ボックスを使用しない場合

```
                        ;    ***************************************
                        ;    *                                     *
                        ;    *        SLOT SELECT 1                *
                        ;    *                                     *
                        ;    ***************************************
                        ;
                        ;    ACC A:       SLOT NO.(0,1)
                        ;
0000'                        ASEG
                             ORG          0F000H
F000    01 FFA0         SLTSEL: LD        BC,0FFA0H
F003    E6 01                AND          01H
F005    F6 02                OR           02H
F007    ED 79                OUT          (C),A
F009    C9                   RET
                             END
```

(2)汎用拡張ボックスを使用した場合

```
                        ;      **********************************
                        ;      *                                *
                        ;      *    SLOT SELECT 2               *
                        ;      *                                *
                        ;      **********************************
                        ;
                        ;    ACC A:    SLOT NO.(0,1) OF CPU
                        ;    REG D:    SLOT NO.(0..3) OF I/O BOX
                        ;
0000'                        ASEG
                             ORG       0F000H
F000    01 FFA0         SLTSL1: LD     BC,0FFA0H
F003    E6 01                AND       01H
F005    F6 02                OR        02H
F007    ED 79                OUT       (C),A
F009    7A                   LD        A,D
F00A    01 FF00              LD        BC,0FF00H
F00D    E6 03                AND       03H
F00F    ED 79                OUT       (C),A
F011    C9                   RET
                             END
```

(3)デバイスコードの読出方法

```
                    ;       *** DEVICE NO. READ ***
                    ;
0000'                       ASEG
                            ORG     0F000H
                    ;
FFA0                CPUSEL  EQU     0FFA0H
FF00                IOSEL   EQU     0FF00H
                    ;
F000    3E FF       DVRD:   LD      A,0FFH
F002    32 F04E             LD      (CPU0),A
F005    01 0009             LD      BC,SLOT7-CPU0   ;SET BYTE COUNT
F008    11 F04F             LD      DE,CPU1         ;SET DES.ADRESS
F00B    21 F04E             LD      HL,CPU0         ;SET SOURCE ADRESS
F00E    ED B0               LDIR                    ;BLOCK TRANSFER
                    ;
F010    DD 21 F04E  LD      IX,CPU0                 ;SET CPU SLOT DATA ADRESS
F014    FD 21 F050          LD      IY,SLOT0        ;SET IO SLOT DATA ADRESS
F018    16 00               LD      D,0
F01A    7A          DVLOP:  LD      A,D             ;SET CPU SLOT NO.
F01B    F6 02               OR      02H             ;SET RAM SELECT FLAG
F01D    01 FFA0             LD      BC,CPUSEL       ;SET IO PORT ADRESS
F020    ED 79               OUT     (C),A           ;SELECT CPU SLOT
```

（次ページに続く）

```
F022    1E 00                  LD      E,0
F024    01 FF00                LD      BC,IOSEL        ;SET IO PORT ADRESS
F027    7B             DVLOP1: LD      A,E             ;SET IO BOX SLOT NO.
F028    ED 79                  OUT     (C),A           ;SELECT IO BOX SLOT
F02A    ED 78                  IN      A,(C)           ;READ DEVICE NUMBER
F02C    67                     LD      H,A
F02D    3E 04                  LD      A,4             ;SET IO BOX DESELECT NO.
F02F    ED 79                  OUT     (C),A           ;IO BOX DESELECT
F031    ED 78                  IN      A,(C)           ;READ DEVICE NO.
F033    BC                     CP      H
F034    20 03          JR      NZ,DRNEQ                ;IF CPU=IOBOX
F036    DD 77 00               LD      (IX),A          ;     THEN SET CPU DEVICE NO.
F039    7C             DRNEQ:  LD      A,H
F03A    FD 77 00               LD      (IY),A          ;STORE DEVICE NO.
F03D    FD 23                  INC     IY
F03F    1C                     INC     E
F040    7B                     LD      A,E
F041    FE 04                  CP      04H
F043    20 E2                  JR      NZ,DVLOP1       ;IO BOX SLOT END ?
F045    DD 23                  INC     IX
F047    14                     INC     D
F048    7A                     LD      A,D
F049    FE 02                  CP      02H
F04B    20 CD                  JR      NZ,DVLOP        ;CPU SLOT END ?
F04D    C9                     RET
                       ;
                       ;       CPUn
                       ;            0FFH       (IO BOX) OR (NO DEVICE)
                       ;            OTHER      DEVICE No.(Direct set)
                       ;       CPUn OR SLOTn
                       ;            0FFH       NO DEVICE
                       ;            OTHER      DEVICE NO
F04E    01             CPU0:   DB      1               ;cpu slot #1
F04F    01             CPU1:   DB      1               ;cpu slot #2
F050    01             SLOT0:  DB      1               ;IO SLOT #0
F051    01             SLOT1:  DB      1               ;IO SLOT #1
F052    01             SLOT2:  DB      1               ;IO SLOT #2
F053    01             SLOT3:  DB      1               ;IO SLOT #3
F054    01             SLOT4:  DB      1               ;IO SLOT #4
F055    01             SLOT5:  DB      1               ;IO SLOT #5
F056    01             SLOT6:  DB      1               ;IO SLOT #6
F057    01             SLOT7:  DB      1               ;IO SLOT #7
                       ;
                               END


Symbols:
CPU0    F04E    CPU1    F04F    CPUSEL  FFA0    DRNEQ   F039
DVLOP   F01A    DVLOP1  F027    DVRD    F000    IOSEL   FF00
SLOT0   F050    SLOT1   F051    SLOT2   F052    SLOT3   F053
SLOT4   F054    SLOT5   F055    SLOT6   F056    SLOT7   F057
```

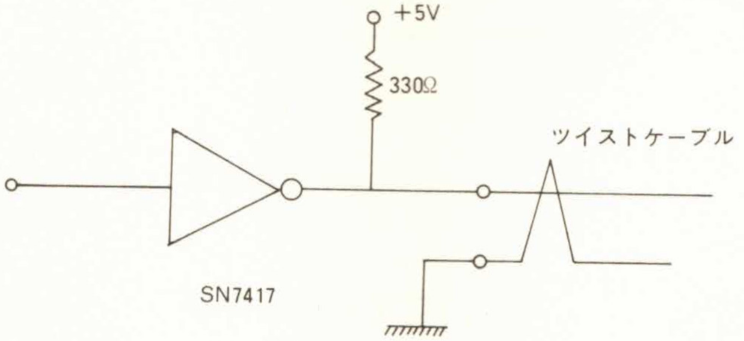
§ 4. インタフェイス

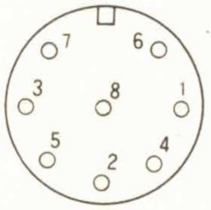
4-1　カラーモニタ・インタフェイス

番号		端　子　名
1		＋12V
2		GND
3		空
4	出	$\overline{\text{HSYNC}}$
5	出	$\overline{\text{VSYNC}}$
6	出	RED
7	出	GREEN
8	出	BLUE
ケース		FG

●RGB同期信号分離方式

注）カラーモニタインタフェイスコネクター
1pin＋12v
グリーンモニタインタフェイスコネクター
1pin＋12v
} トータル電流(0.1max)





DIN8pin

●＋12V　　＋12V電源端子です。RFモジューレータ(FP-1036RF)がこの電源を使用します。

●GND　　信号のグランドです。

●$\overline{\text{HSYNC}}$　　水平同期信号。TTLレベル，負論理の信号です。

●$\overline{\text{VSYNC}}$　　垂直同期信号。TTLレベル，負論理の信号です。

●RED　　赤のビデオ信号。TTLレベル，正論理の信号です。

●GREEN　　　　緑のビデオ信号。TTLレベル，正論理の信号です。

●BLUE　　　　青のビデオ信号。TTLレベル，正論理の信号です。

●FG　　　　　フレーム・グランド。

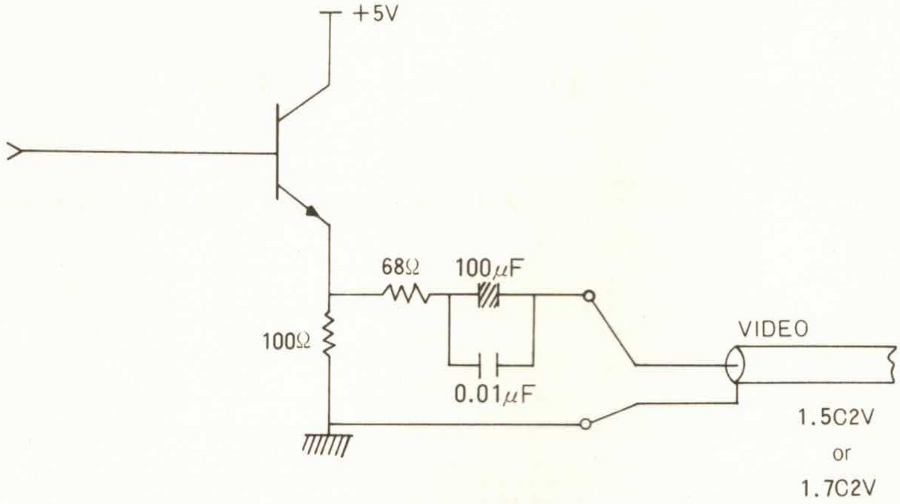
注）コネクター内部で、FGとGNDを接続して使用してください。

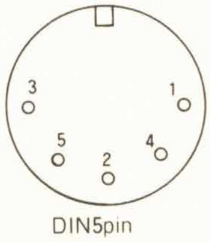
4-2 グリーンモニタ・インタフェイス

番号		端 子 名
1		＋12V
2		GND
3	出	VIDEO
4		空
5	入	$\overline{\text{LPS}}$(ライトペン)
ケース		FG

●コンポジット信号

出力インピーダンス	75Ω
出力電圧	1Vp-p
ビデオ信号：同期信号	7：3
極性	ビデオ信号が正極性





●＋12V　　電源端子です。

●GND　　信号のグランドです。

●VIDEO　　映像信号と同期信号から合成したコンポジットビデオ信号です。

●$\overline{\text{LPS}}$　　ライトペン・ストローブ信号。

●FG　　フレーム・グランド。

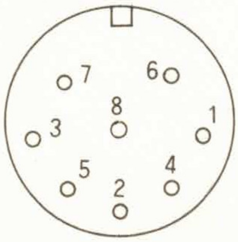
注）コネクター内部で、FGとGNDを接続して使用してください。

4-3　カセットテープレコーダ・インタフェース

番号		端　子　名
1		+5V
2		GND
3		空
4	出	MIC
5	入	EAR
6		REM+
7		REM−
8		空
ケース		FG

MIC端子	出力インピーダンス	5KΩ
	出力電圧	$3mV_{p\text{-}p}$
EAR端子	入力インピーダンス	10KΩ
	入力電圧	$3\sim10V_{p\text{-}p}$
REMORT	定格	24V　1A

注）カセットテープレコーダ
インタフェイスコネクター1pin
プリンタインタフェイスコネクター35pin
} +5Vトータル電流（0.1Amax）



●+5V　　電源端子です。

●GND　　信号のグランドです。

●MIC（赤）　　カセットテープレコーダへの出力信号です。

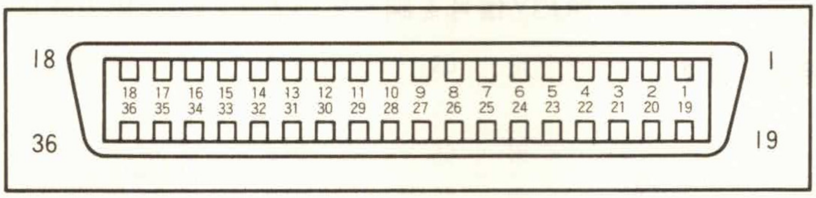
●EAR（白）　　カセットテープレコーダからの入力信号です。

●REM（+）
REM（−）　　カセットテープレコーダのリモート端子をつなぎます。
本体では、この２本の線をリレーによって断続しています。

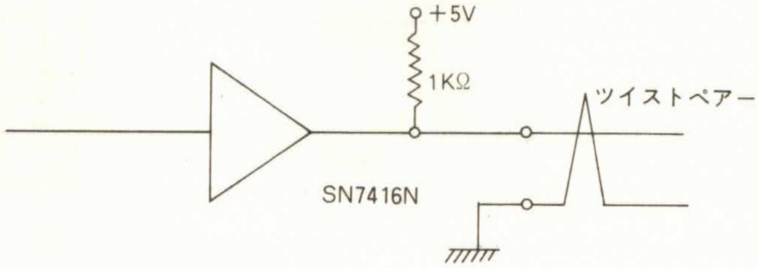
●FG　　フレームグランド

4-4 プリンタ・インタフェイス

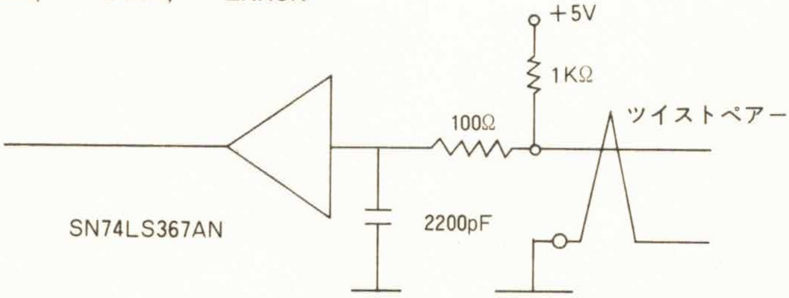
番号		端 子 名	番号		端 子 名
1	出	$\overline{\text{STROBE}}$	19		GND
2	出	DATA1	20		GND
3	出	DATA2	21		GND
4	出	DATA3	22		GND
5	出	DATA4	23		GND
6	出	DATA5	24		GND
7	出	DATA6	25		GND
8	出	DATA7	26		GND
9	出	DATA8	27		GND
10	入	$\overline{\text{ACKNLG}}$	28		GND
11	入	$\overline{\text{BUSY}}$	29		GND
12		空	30		GND
13		空	31	出	$\overline{\text{INIT}}$
14		空	32	入	$\overline{\text{ERROR}}$
15		空	33		GND
16		OV	34		空
17		FG	35		+5V
18		空	36		空



DATA1〜8, $\overline{\text{STROBE}}$ $\overline{\text{INIT}}$



$\overline{\text{ACKNLG}}$, BUSY, $\overline{\text{ERROR}}$



●DATA 7〜8　　正論理のデータ信号。JIS8ビットコードを出力します。MSBはDATA 8です。

●$\overline{\text{STROBE}}$　　負論理のデータ信号。プリンタへ出力するDATA 1〜8が有効であることを示します。

●$\overline{\text{ACKNLG}}$　　プリンタからのデータ受付完了パルス。

●$\overline{\text{BUSY}}$　　プリンタからの動作中を示す信号。この信号がHivベルのとき、データの出力はしません。

●$\overline{\text{INIT}}$　　プリンタに対するイニシャル信号。電源投入時、または、FP-1000／1100がプリンタの応答待ちをしているとき、BREAKキーを操作すると出力します。

約100μSの巾で出力します。

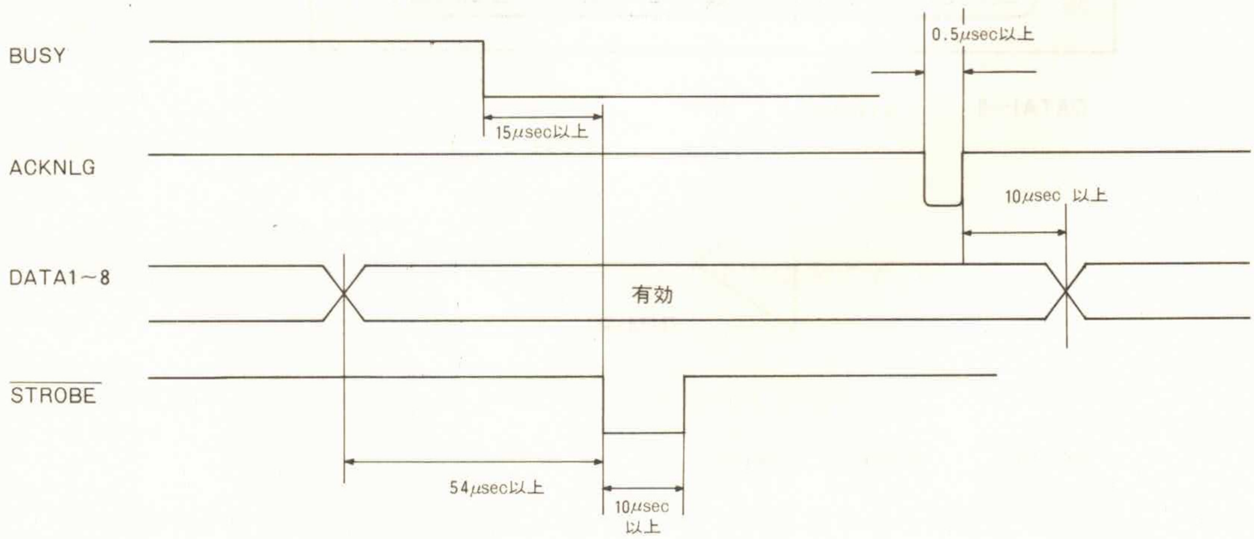
●$\overline{\text{ERROR}}$　　プリンタからのエラー信号。この信号によりNRエラーとなります。

●GND　　信号のツイストペアリターン用のグランド。

●0V　　信号のグランドです。

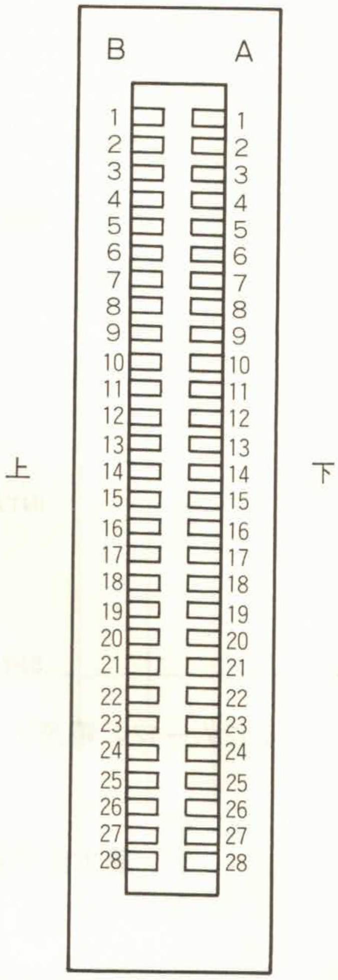
●FG　　フレームグランド。

プリンタへ出力する前に必ず、BUSY信号を調べています。また、出力後は、$\overline{\text{ACKNG}}$信号の応答を確認します。

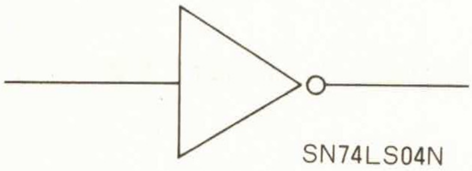


4-6 汎用拡張スロットインタフェース

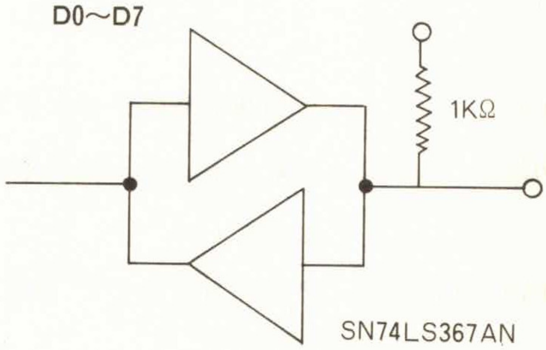
番号		端子名	番号		端子名
1		+5V	1		GND
2		+5V	2		GND
3		+12V	3		+12V
4	出	$\overline{PSL}$	4	出	φ(3.9936MHz)
5	出	$\overline{BSL}$	5	出	$\overline{RESET}$
6	出	$\overline{REQ}$	6	出	$\overline{REQD}$
7	出	$\overline{RD}$	7	出	$\overline{WR}$
8	出	$\overline{CS}$	8		
9			9		
10	入	$\overline{NMI}$	10	入	$\overline{LPS}$
11	入	$\overline{INTA}$	11	入	$\overline{INTB}$
12	入	$\overline{INTC}$	12	入	$\overline{INTD}$
13	入出	D0	13	入出	D1
14	入出	D2	14	入出	D3
15	入出	D4	15	入出	D5
16	入出	D6	16	入出	D7
17	出	$\overline{A0}$	17	出	$\overline{A1}$
18	出	$\overline{A2}$	18	出	$\overline{A3}$
19	出	$\overline{A4}$	19	出	$\overline{A5}$
20	出	$\overline{A6}$	20	出	$\overline{A7}$
21	出	$\overline{A8}$	21	出	$\overline{A9}$
22	出	$\overline{A10}$	22	出	$\overline{A11}$
23	出	$\overline{A12}$	23	出	$\overline{A13}$
24	出	$\overline{A14}$	24	出	$\overline{A15}$
25		−12V	25		−5V
26		+5V	26		GND
27		+5V	27		GND
28			28		

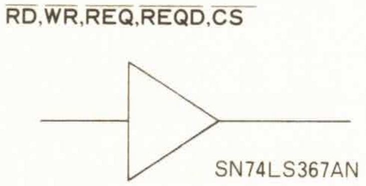


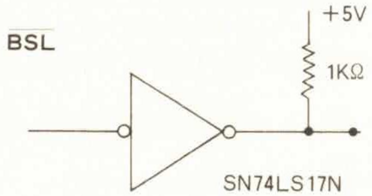
$\overline{A0}$〜$\overline{A15}$, $\overline{PSL}$

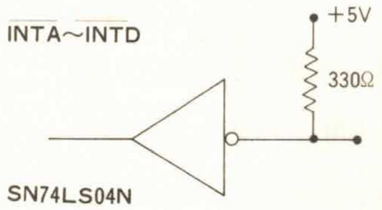


D0〜D7







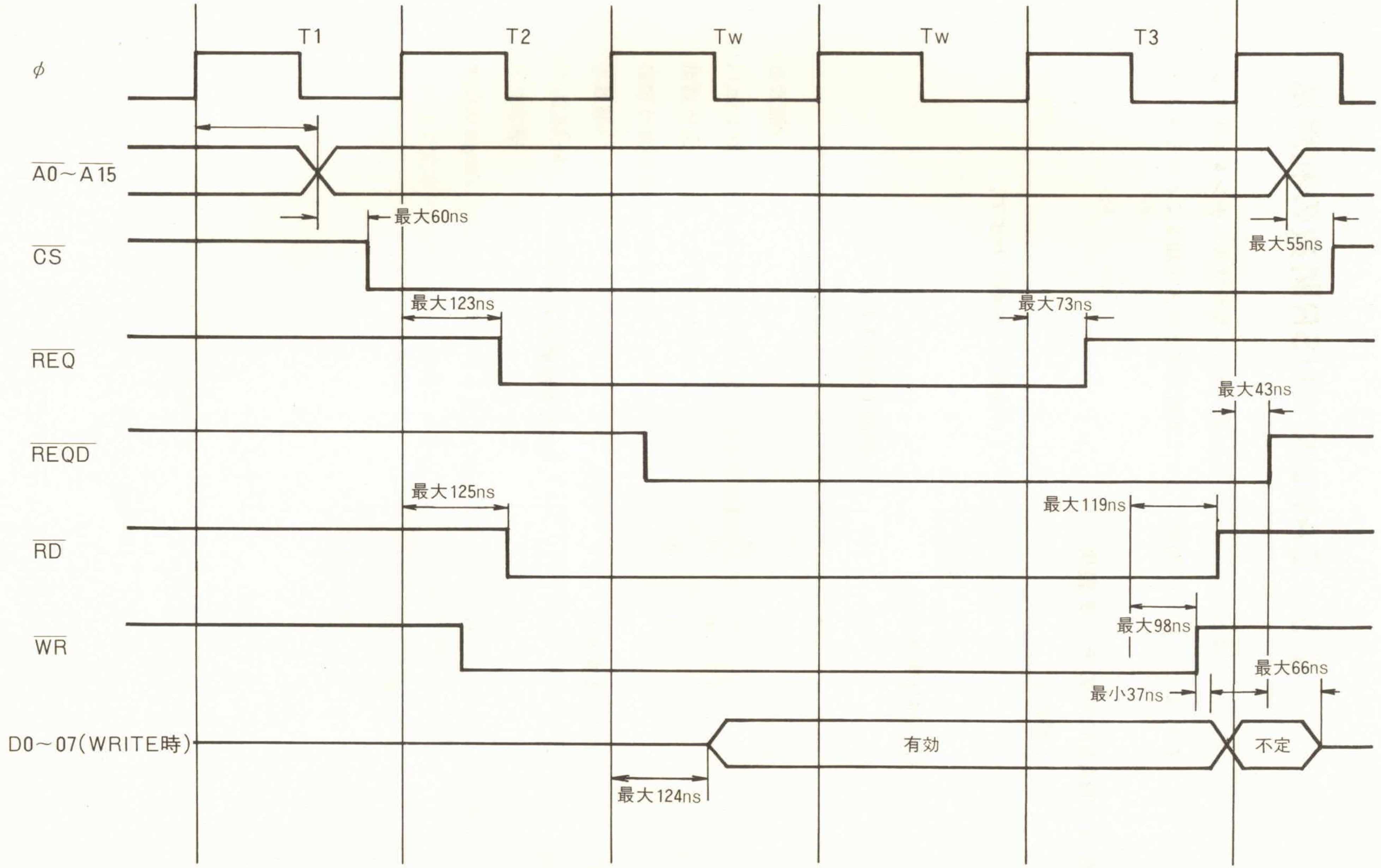


●5V,−5,12V,−12V 電源です。

最大出力電流
- +12V　0.1Amax
- +5V　1.0Amax
- −5V, −12V　トータル0.1Amax

●$\overline{PSL}$, $\overline{BSL}$	スロット選択信号。"Low"でそのスロットが選択されていることを示します。$\overline{BSL}$と$\overline{PSL}$は出力信号のレベルが違います。
●$\overline{REQ}$, $\overline{REQD}$	CPUの$\overline{IOREQ}$信号です。$\overline{REQD}$は1クロック送らせた信号です。
●$\overline{RD}$, $\overline{WR}$	CPUの$\overline{RD}$, $\overline{WR}$信号です。$\overline{REQ}$または$\overline{REQD}$とANDをとって使用します。
●$\overline{NMI}$	CPUの$\overline{NMI}$信号です。
●$\overline{INTA}$~$\overline{INTD}$	インタラプト信号です。3-3-1インタラプトレジスタを参照して下さい。
●ϕ	3.9936MHzのクロックです。
●$\overline{CS}$	スロットにあるデバイスに対して、デバイスコードの出力を要求します。
●$\overline{A0}$~$\overline{A15}$	アドレス出力です。使用可能なアドレスは0000H~0FEFFHまでです。
●D0~D7	データ入出信号です。

汎用拡張スロットでの主要なタイミング



§ 5. CALL文・USR関数の利用法

CALL,USR関数によって、C82-BASICでマシン語サブルーチンを利用することができます。また、VARPTR関数によって変数のアドレスを知ることができます。

5-1　BASIC内のデータ表現

(1)整数型

2byte

下位 8 ビット
下位 8 ビット

2 の補数型で格納されます。

(2)単精度実数型

6byte

符号	e1
e2	d1
d2	d3
d4	d5
d6	d7
d8	d9

BCDで格納されます。

(3)倍精度実数型

11byte

符号	e1
e2	e1
d2	d3
d16	d17
d18	d19

BCDで格納されます。

(4)倍々精度実数型

16byte

符号	e1
e2	e1
e2	e3
	e
d26	e27
d28	e29

BCDで格納されます。

〈符号〉

	仮数	指数
0	+,0	−
1	+	+,0
5	−	−
6	−	+,0

〈指数部〉

e1,e2で指数2桁をBCDで表わし、00～99までの数が入ります。

指数が正の場合、そのままの値となります。

指数が負の場合、100の補数をとります。

〈仮数部〉

d1,d2,d3,d4………の並びで仮数を表わします。d1とd2の間に、小数点があります。

(5)文字型

先頭の1byteに文字列の長さが入ります。2byte目以降が実際の長さです。データは、この形で文字データエリアに確保されています。

5-2　CALL文

CALL文では、引数をレジスターに渡すことができます。また。実際にマシン語のCALL命令でサブルーチンコールをしていますから、RET命令で復帰することができます。

```
10 CLEAR 1023,&HEFFF
20 GOSUB 100
30 SCREEN 0
40 WIDTH 80
50 LOCATE 0,0
60 INPUT "COLOR=",C
70 C=C MOD 8
80 CALL &HF000,C,0,0
90 GOTO 60
100 DATA 32,D2,9F
110 DATA AF
120 DATA 32,D3,9F
130 DATA ED,53,CA,9F
140 DATA 22,CC,9F
150 DATA CD,03,07
160 DATA CD,0F,07
170 DATA 28,FB
180 DATA FE,1C
190 DATA 20,06
200 DATA CD,43,F0
210 DATA C3,11,F0
220 DATA FE,1D
230 DATA 20,06
240 DATA CD,58,F0
250 DATA C3,11,F0
260 DATA FE,1E
270 DATA 20,06
280 DATA CD,88,F0
290 DATA C3,11,F0
300 DATA FE,1F
310 DATA 20,06
320 DATA CD,73,F0
330 DATA C3,11,F0
340 DATA FE,0D
350 DATA 20,CF
360 DATA C9
370 REM
380 DATA 2A,CA,9F
390 DATA 23,37,3F,11,7F,02
400 DATA ED,52
410 DATA 30,01
420 DATA 19
430 DATA 22,CA,9F
440 DATA CD,A3,F0
450 DATA C9
460 REM
470 DATA 2A,CA,9F
480 DATA 2B
490 DATA E5
500 DATA 11,FF,FF
510 DATA 37
520 DATA 3F
530 DATA ED,52
540 DATA 20,05
550 DATA E1
560 DATA 21,7F,02
570 DATA E5
580 DATA E1
590 DATA 22,CA,9F
600 DATA CD,A3,F0
610 DATA C9
620 REM
630 DATA 2A,CC,9F
640 DATA 23
650 DATA 37,3F
660 DATA 11,C7,00
670 DATA ED,52
680 DATA 30,01
690 DATA 19
700 DATA 22,CC,9F
710 DATA CD,A3,F0
720 DATA C9
730 REM
740 DATA 2A,CC,9F
750 DATA 2B
760 DATA E5
770 DATA 11,FF,FF
780 DATA 37
790 DATA 3F
800 DATA ED,52
810 DATA 20,05
820 DATA E1
830 DATA 21,C7,00
840 DATA E5
850 DATA E1
860 DATA 22,CC,9F
870 DATA CD,A3,F0
880 DATA C9
890 REM
900 DATA CD,BC,07
910 DATA CD,A2,08
920 DATA C9
930 DATA @
940 FOR I=&HF000 TO &HFFFF:READ A$:IF A$="@" THEN RETURN
950 POKE I,VAL("&H"+A$)
960 NEXT
```

```
                        ;       **** GRAPHIC PAINTER ****
                        ;
                        ;       A:      COLOR(0-7)
                        ;       DE:     X
                        ;       HL:     Y
                        ;
0000'                           ASEG
                                ORG     0F000H
07BC                    XGCOL   EQU     07BCH           ;SET COLOR ENTRY POINT
9FD2                    COLOR   EQU     9FD2H           ;COLOR PARAM FOR XGCOL
08A2                    XPLOT   EQU     08A2H           ;PLOT ENTRY POINT
9FCA                    X       EQU     9FCAH           ;COR-X FOR PLOT
9FCC                    Y       EQU     9FCCH           ;COR-Y FOR PLOT
0703                    KRNGC   EQU     0703H           ;KEY BUFF CLEAR
070F                    GETRG   EQU     070FH           ;KEY BUFF READ
                        ;       A       DATA
                        ;       Z       NO DATA
                        ;
F000    32 9FD2         GP:     LD      (COLOR),A
F003    AF                      XOR     A
F004    32 9FD3                 LD      (COLOR+1),A
F007    ED 53 9FCA              LD      (X),DE
F00B    22 9FCC                 LD      (Y),HL
F00E    CD 0703                 CALL    KRNGC
F011    CD 070F         MAIN:   CALL    GETRG
F014    28 FB                   JR      Z,MAIN
F016    FE 1C                   CP      1CH
F018    20 06                   JR      NZ,MAIN1
F01A    CD F043                 CALL    RIGHT
F01D    C3 F011                 JP      MAIN
F020    FE 1D           MAIN1:  CP      1DH
F022    20 06                   JR      NZ,MAIN2
F024    CD F058                 CALL    LEFT
F027    C3 F011                 JP      MAIN
F02A    FE 1E           MAIN2:  CP      1EH
F02C    20 06                   JR      NZ,MAIN3
F02E    CD F088                 CALL    UP
F031    C3 F011                 JP      MAIN
F034    FE 1F           MAIN3:  CP      1FH
F036    20 06                   JR      NZ,MAIN4
F038    CD F073                 CALL    DOWN
F03B    C3 F011                 JP      MAIN
F03E    FE 0D           MAIN4:  CP      0DH
F040    20 CF                   JR      NZ,MAIN
F042    C9                      RET
                        ;
F043    2A 9FCA         RIGHT:  LD      HL,(X)
F046    23                      INC     HL
F047    37                      SCF
F048    3F                      CCF
F049    11 027F                 LD      DE,639
F04C    ED 52                   SBC     HL,DE
F04E    30 01                   JR      NC,RIGHT1
F050    19                      ADD     HL,DE
F051    22 9FCA         RIGHT1: LD      (X),HL
F054    CD F0A3                 CALL    ACT
```

（次ページに続く）

```
F057   C9                RET
                  ;
F058   2A 9FCA      LEFT:  LD    HL,(X)
F05B   2B                DEC   HL
F05C   E5                PUSH  HL
F05D   11 FFFF           LD    DE,0FFFFH
F060   37                SCF
F061   3F                CCF
F062   ED 52             SBC   HL,DE
F064   20 05             JR    NZ,LEFT1
F066   E1                POP   HL
F067   21 027F           LD    HL,639
F06A   E5                PUSH  HL
F06B   E1           LEFT1: POP   HL
F06C   22 9FCA           LD    (X),HL
F06F   CD F0A3           CALL  ACT
F072   C9                RET
                  ;
F073   2A 9FCC      DOWN:  LD    HL,(Y)
F076   23                INC   HL
F077   37                SCF
F078   3F                CCF
F079   11 00C7           LD    DE,199
F07C   ED 52             SBC   HL,DE
F07E   30 01             JR    NC,DOWN1
F080   19                ADD   HL,DE
F081   22 9FCC      DOWN1: LD    (Y),HL
F084   CD F0A3           CALL  ACT
F087   C9                RET
                  ;
F088   2A 9FCC      UP:    LD    HL,(Y)
F08B   2B                DEC   HL
F08C   E5                PUSH  HL
F08D   11 FFFF           LD    DE,0FFFFH
F090   37                SCF
F091   3F                CCF
F092   ED 52             SBC   HL,DE
F094   20 05             JR    NZ,UP1
F096   E1                POP   HL
F097   21 00C7           LD    HL,199
F09A   E5                PUSH  HL
F09B   E1           UP1:   POP   HL
F09C   22 9FCC           LD    (Y),HL
F09F   CD F0A3           CALL  ACT
F0A2   C9                RET
                  ;
F0A3   CD 07BC      ACT:   CALL  XGCOL
F0A6   CD 08A2           CALL  XPLOT
F0A9   C9                RET
                         END


Symbols:
ACT    F0A3   COLOR  9FD2   DOWN   F073   DOWN1  F081
GETRG  070F   GP     F000   KRNGC  0703   LEFT   F058
LEFT1  F06B   MAIN   F011   MAIN1  F020   MAIN2  F02A
MAIN3  F034   MAIN4  F03E   RIGHT  F043   RIGHT1 F051
UP     F088   UP1    F09B   X      9FCA   XGCOL  07BC
XPLOT  08A2   Y      9FCC
```

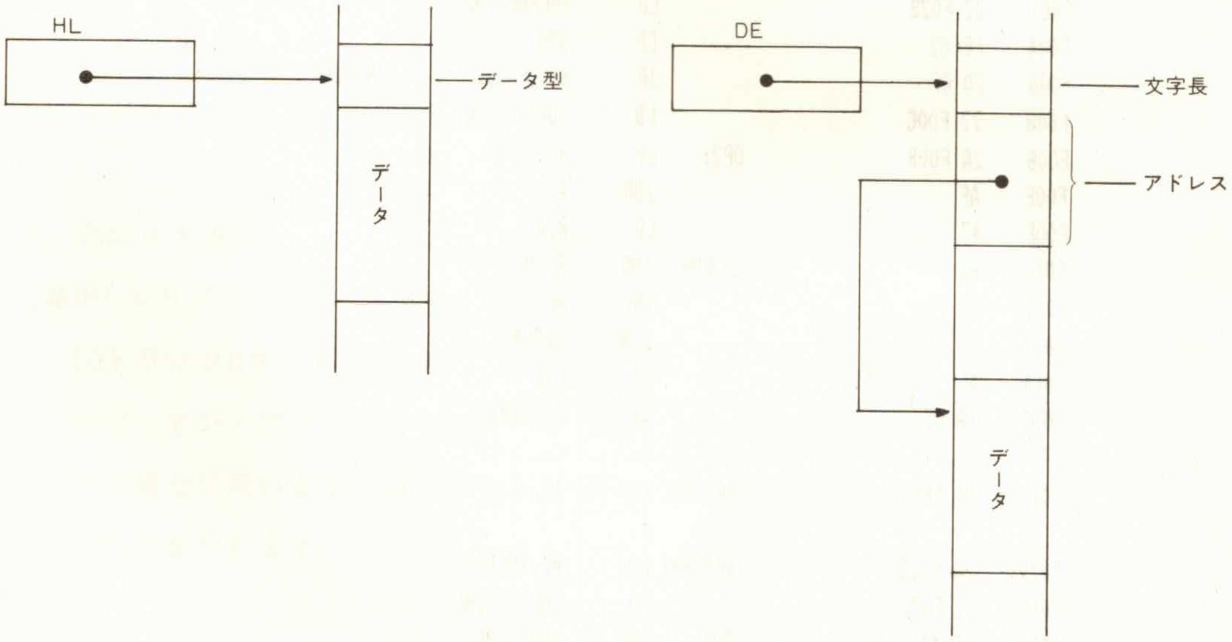
5-3 USR関数

USR関数は、CALL文と同様にマシン語サブルーチンを呼び出す機能を持ちますが、引数の渡し方、また、マシン語ルーチンから渡された値を関数値とする点で異ります。

(1)USR関数からマシン語サブルーチンへの引数渡し、

データの型	Aレジスタ	
	00	文字型
	02	整数型
	06	単精度実数型
	11	倍精度実数型
	16	倍々精度実数型

データの位置			
	数値の場合。	(HL)番地	データの型
		(HL＋1)番地	データ
	文字の場合。	(DE)番地	文字長
		(DE＋1)番地	データのアドレス



(2)マシン語サブルーチンからUSR関数への結果渡し。

(1)と同じ場所へ同じ形で渡します。また、HL,DEも同様にセットします。

(3)使用例

引数から始まる256byteの合計を256で除した余りを値とする関数です。必ず、引数は整数型でなければなりません。また、結果のデータ型も整数です。

```
10 CLEAR 1023,&HEFFF
20 GOSUB 90
30 DEFINT A-Z
40 DEFUSR=&HF000
50 CLS
60 INPUT "START ADRESS=",ADR
70 PRINT USR(ADR)
80 GOTO 60
90 DATA 23,22,28,F0,FE,02,20,0F,22,0C,F0,2A,0B,F0,AF,47,86,23
100 DATA 10,FC,4F,18,03,01,FF,FF,2A,28,F0,22,22,F0,ED,43,20,F0,2B,36,02,C9,@
110 FOR I=&HF000 TO &HFFFF:READ A$:IF A$="@" THEN RETURN
120 POKE I,VAL("&H"+A$)
130 NEXT I
140 RETURN
```

```
                                ;       **** CHECK SUM OF 256 BYTE ***
                                ;
0000'                                   ASEG
                                        ORG     0F000H
                                ;
F000    23                      CKSUM:  INC     HL
F001    22 F028                         LD      (HLBAK),HL
F004    FE 02                           CP      02H
F006    20 0F                           JR      NZ,ERROR        ;INTEGER ?
F008    22 F00C                         LD      (OP2+1),HL
F00B    2A F00B                 OP2:    LD      HL,(OP2)
F00E    AF                              XOR     A
F00F    47                              LD      B,A
F010    86                      CKSUML: ADD     A,(HL)
F011    23                              INC     HL
F012    10 FC                           DJNZ    CKSUML
F014    4F                              LD      C,A
F015    18 03                           JR      RETURN
                                ;
F017    01 FFFF                 ERROR:  LD      BC,-1
                                ;
F01A    2A F028                 RETURN: LD      HL,(HLBAK)
F01D    22 F022                         LD      (OP3+2),HL
F020    ED 43 F020              OP3:    LD      (OP3),BC
F024    2B                              DEC     HL
F025    36 02                           LD      (HL),02H
F027    C9                              RET
                                ;
F028    02                      HLBAK:  DB      2
                                        END
```

§ 6. C_{82}-BASICのファイル管理

ファイル管理の対象は、FDD,RAMパック,ROMパックです。これらは全て共通の方法で管理されています。

6-1 FDDのファイル管理

(1)物理的構成

1セクタ　　　　＝256バイト

1トラック　　　＝16セクタ＝4096バイト

1サイド　　　　＝40トラック＝640セクタ＝163840バイト

1ディスクシート＝2サイド　＝80トラック　＝1280セクタ

＝327680バイト

セクタNo.(1～16)

トラックNo.(0～39)

サイドNo.(0，1)

ドライブNo.(0～7)

(2)エリア構成

トラック＼セクタ	サイド0				サイド1			
	1	2	～	16	1	2	～	16
0	IPL	予備						
1	FAT	ディレクトリ						
2～39	データ							

●IPLエリア

(0トラック0サイド1セクタ)

パワーオン・ブート機能を使用する場合に用います。通常は全て00が書かれます。

●予備エリア

(0トラック0サイド2～16セクタ,0トラック1サイド)

使用してません。

●FATエリア

(1トラック0サイド1セクタ)

ディスクシートの使用状況の情報が記録されています。

●ディレクトリエリア

（1トラック0サイド2～16セクタ，1トラック1サイド）

ファイル名，属性が記録されています。152(160ファイル)ファイルまで管理できます。

●データエリア

(2トラック～39トラック)

データが記録されています。容量は、311296バイトあります。

(3)FAT

ディスクシートの各セクタは、クラスタ単位で管理しています。8セクタで1クラスタを構成し、0トラック0サイド1セクタから8セクタづつ順に0から始まる番号がついています。クラスタの番号のふり方は、トラックの小さい方から大きい方へ、同一のトラック内ではサイド0からサイド1へ，またセクタ1からセクタ16へという順にふります。

0～159 クラスタまで、全体で160クラスタあります。

FATはこの各クラスタに対応して1byteのデータを持ち、それによってクラスタの使用状況を表わします。

トラック＼セクタ	0サイド		1サイド	
	1 ～ 8	9 ～ 16	1 ～ 8	9 ～ 16
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15
～				
38	152	153	154	155
39	156	157	158	159

FATの先頭の1byteは常にOAOHになっています。次のbyteから連続する160byteが各クラスタの使用状況を表わします。

各byteの意味は以下の通り決められています。

OFFH	未使用クラスタ
OFEH	管理外クラスタ
OCOH未満	使用中クラスタ。同じファイルの次に続くクラスタの番号になっています。
OC1H ～ OC8H	使用中クラスタ。ファイルの最後のクラスタであることを表わしています。値からOCOHを引いた値がクラスタ中の最後の使用セクタを表わします。

⑷ディレクトリ

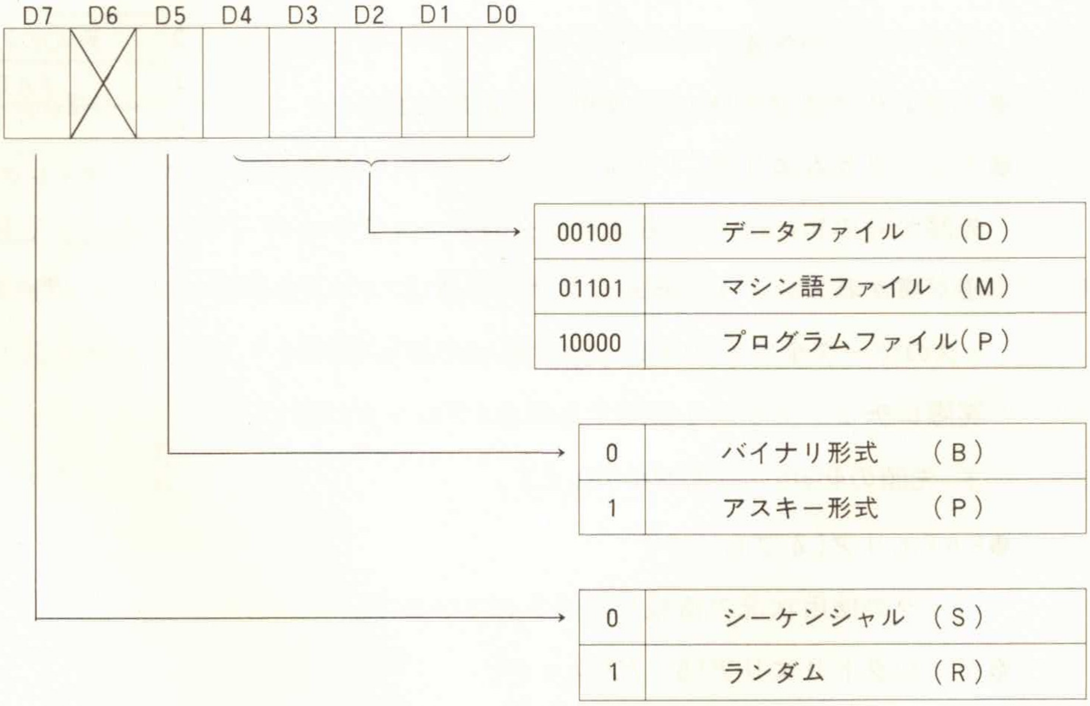
1つのファイルに対するディレクトリの大きさは32byteです。従って、1セクタには8つのディレクトリがあり、ディレクトリエリアには160のディレクトリまで記録されます。

ディレクトリの構成

1	2–9	10–12	13–20	21–22	23–24	25–26	27–28	29	30	31	32
①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫

①ファイル識別子(1byte)

bit構成によりファイルの属性を表わします。



②ファイル名

③拡張子

④システムパラメータ領域

FFHコードに初期化します

⑤マシン語ファイル先頭アドレス

⑥マシン語ファイル終了アドレス

⑦マシン語ファイル実行先頭番地

⑧プログラムバイト数

⑨クラスタエントリー

⑩属性

SET文で設定される属性です。

OFOH————ライトプロテクト("P"属性)

OFH ————リード・アフター・ライト("R"属性)

⑪00Hに設定されます。

⑫00Hに設定されます。

6-2 RAM·ROMパックのファイル管理

ファイル管理の方法は基本的にFDDと同じ方法で管理しています。

(1)物理的構成

アドレスの先頭から256byte単位で、0から始まるアドレスを振り、それをブロック番号としています。

1ブロック＝256byte

パック全体の容量は、そのハード的な容量によります。

(2)エリア構成

●IPLエリア(0ブロック)

パワーオンブート機能を使用する場合に用います。

通常は全て00が書かれます。

●予備エリア(1,2ブロック)使用してません。

●チェックサムエリア(3ブロック)

先頭からの1byte単位で各ブロックのチェックサムの値が書かれています。チェックサムの値は、そのブロックのデータ全てを加算し、256で割った余りです。

実際にチェックサムを使用するのは4ブロック以降です。先頭の4byteには00がかかれます。

●FATエリア(4ブロック)

パックの使用状況の情報が記録されています。

●ディレクトリエリア(5～7ブロック)

ファイル名、属性が記録されています。24ファイルまで管理できます。

●データエリア(8ブロック以降)

データが記録されています。RAMパック，ROMパックは最大32kbyteまで管理できる構成となっています。但しメモリの実装によってデータ量は変化します。RAMパックの場合は14336byteです。

(3)FAT

RAM／ROMパックの各ブロックはクラスタ単位で管理しています。8セクタで1クラスタを構成し、先頭から順に0から始まる番号がついています。

RAMパックの場合には0～7クラスタで合計8クラスタ、ROMパックの場合は0～15クラスタで合計16クラスタあります。

FATはこの各クラスタに対応して1byteのデータを持ち、それによってクラスタの使用状況を示します。

FATの先頭の1byteはRAMパックの場合 8, ROMパックの場合16とします。次のバイトから始まる8byte(ROMの場合は16byte)が各クラスタの状況を表わします。

各byteの意味は以下の通り決められています。

0FFH	未使用クラスタ
0FEH	管理外クラスタ。ROMパックではROMの実装されていないクラスタも0FEHに設定します。
0C0H未満	使用中クラスタ。同じファイルの次に続くクラスタの番号になっています。
0C1H ～ 0C8H	使用中クラスタ。ファイルの最後のクラスタであることを表わしています。値から0C0Hを引いた値がクラスタ中の最後の使用セクタを表わしています。

⑷ディレクトリ

FDDと全く同じ管理をしています。但し、ディレクトリは最大24まで登録できます。FDDのディレクトリを参照して下さい。

§7．パワーオンブート

FP-1000／1100は電源投入時に回路のイニシャル動作終了後、RAM／ROMパックまたは、FDDの内容を調べ、パワーオンブート指定があった場合にはC_{82}-BASICを放棄して、RAM／ROMパックまたは、FDDの内容をRAMに読み込み、それを実行します。

パワーオンブートのチェックは、以下の順で各デバイスに対して行います。

1，ROMパック　　0から順に7まで

2，FDD　　　　　FDDのうち一番若いスロットにある0番のドライブ

3，RAMパック　　0から順に7まで

パワーオンブートの処理は、RAM／ROMパック，FDDに対して共通です。

1，IPLエリアの内容をチェックします。先頭の1byteが0C3Hでかつ、IPLエリアの総和が256の倍数のとき、パワーオンブートと判断します。

2，IPLエリアの内容を9000Hから90FFHへ読みます。

3，9000Hをサブルーチンコールします。

§8. C_{82}-BASICのエントリーポイント

C_{82}-BASIC内のサブルーチンのエントリーポイントを紹介します。

8-1　キー入力

①KRNGC(0703H)

（機能）　キー入力バッファをクリアします。

（IN条件）　なし

（OUT条件）　なし

②GETRG(070FH)

（機能）　キー入力バッファからデータを取出します。

（IN条件）　なし

（OUT条件）　●Areg　キーデータ(JIS8bitコード)

●Z flag　{ 0　データ有り
　　　　　{ 1　データ無し

8-2　CRT表示

①PRCRT(074FH)

（機能）　Aregの内容をCRTに表示します。

（IN条件）　●Areg　表示コード

（OUT条件）　なし

（説明）　20H～7EH,80H～0FEH　文字表示をします。

00H～1FH,7FH,0FFH　以下のもの以外は無機能。

01H	ROLL	UP(40chモード時)物理的一行のロールアップ動作。
07H	BEEP	20msecのビープ音。
08H	BACK SPACE	カーソルを一文字戻し、そこにスペースをを書きます。
0BH	HOME	
0CH	CLS	画面消去
0DH	CRLF	復帰改行。最下行ではスクロール動作。
10H	ROLL	DOWN(40CHモード時)物理的一行のロールダウン。
1CH	(→)	カーソル移動。
1DH	(←)	カーソル移動。
1EH	(↑)	カーソル移動。
1FH	(↓)	カーソル移動。

●スクリーンバッファへの書込は行いません。

●C_{82}-BASIC内のカーソル位置には影響を与えません。

②CTPTN(0753H)

（機能）　Aregの内容をCRTに表示します。

（IN条件）　●Areg　表示コード

（OUT条件）　なし

（説明）　20H～0FFH　「PRCRT」と動作は同じです。

00H～1FH　コントロール動作をせずに、シンボル文字を表示します。この文字はKEYLIST でコントロールコード表示をしたときに現われる文字と同じものです。

例　0DH→$^{C}{}_{R}$

0BH→$^{H}{}_{M}$

39

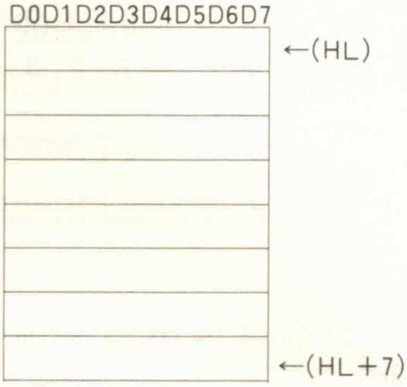
③XOT08(077EH)

（機能）　HLregで指したアドレスから連続した8byteを文字パターンとしてCRTに表示します。

（IN条件）　●HLreg　パターン・アドレス

（OUT条件）　なし

（説明）　パターンとデータの関係は下図の通り。



④XCSR(07A4H)

（機能）　カーソル位置を移動します。

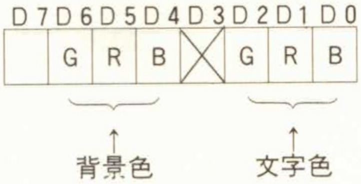
（IN条件）　9FD7H　X座標

9FD8H　Y座標

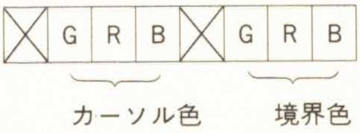
⑤XCOLOR(07B0H)

（機能）　サブCPUに色を指示します。(COLOR文に相等)

（IN条件）　9FD5H



9FD6H



⑥XREV(07DBH)

（機能）　リバース指定をします。

（IN条件）　なし

（OUT条件）　なし

⑦XNORM(07E0H)

（機能）　　ノーマル指定をします。

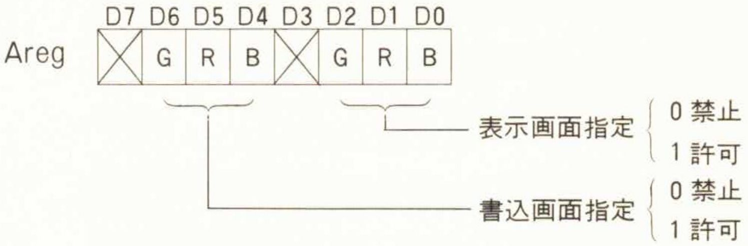
（IN条件）　なし

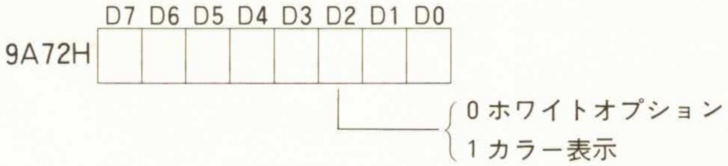
（OUT条件）　なし

⑧XRWIH(07E5H)

（機能）　　スクリーンモード2での表示画面,書き込み画面を指定します。

（IN条件）





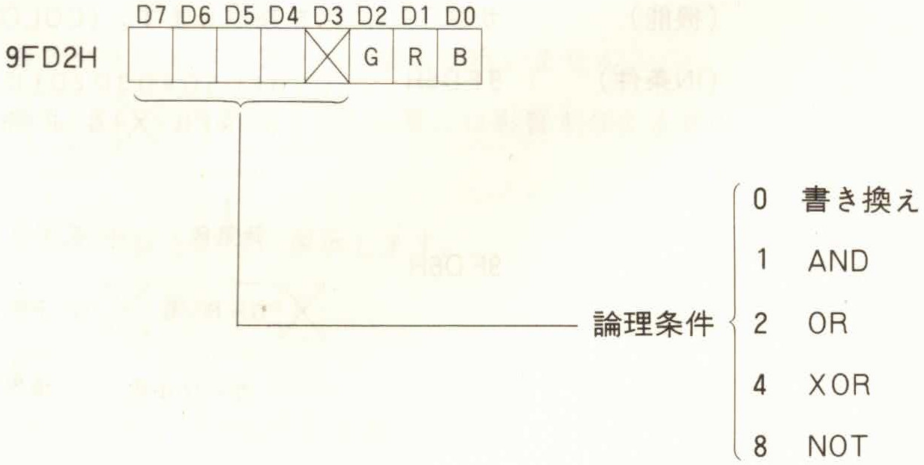
※9A72Hの他のbitは変化させてはいけません。

（OUT条件）　なし

⑨XGCOL(07BCH)

（機能）　　XLINEルーチン,XPLOTルーチンを呼ぶ前に使用し、その機能を指定します。

（IN条件）



（OUT条件）　なし

⑩LTYP(07F9H)

（機能）　　XLINEルーチンを呼ぶ前に使用し、直線のパターンを指定します。DRAW文のラインタイプと同じです。

（IN条件）　0A02FH（下位）｝ラインタイプ
　　　　　　0A030H（上位）｝

（OUT条件）　なし

⑪XLINE（0828H）

（機能）　直線を引きます。色と論理条件は、XGCOLルーチンで指定れます。ラインタイプはLTYPルーチンで指定されます。

（IN条件）　9FCAH（下位）｝始点のX座標
　　　　　　9FCBH（上位）｝
　　　　　　9FCCH（下位）｝始点のY座標
　　　　　　9FCDH（上位）｝
　　　　　　9FCEH（下位）｝終点のX座標
　　　　　　9FCFH（上位）｝
　　　　　　9FD0H（下位）｝終点のY座標
　　　　　　9FD1H（上位）｝

（OUT条件）　なし

⑫XPLOT（08A2H）

（機能）　指定した位置に点を打ちます。色と論理条件はXGCOLルーチンで指定されます。

（IN条件）　●9FCAH（下位）｝X座標
　　　　　　9FCBH（上位）｝
　　　　　　●9FCCH（下位）｝Y座標
　　　　　　9FCDH（上位）｝

（OUT条件）　なし

⑬XRCOL（08EFH）

（機能）　指定した点の色のコードを読みます。

（IN条件）　DEreg X座標
　　　　　　HLreg Y座標

（OUT条件）　Areg　色コード（0～7）

8-3　プリンタ出力

①PRLPT(079BH)

(機能)　サブCPUへプリントデータを渡します。

(IN条件)　Areg　プリントデータ

(OUT条件)　なし

(説明)　プリントデータをサブCPUへ渡すだけのルーチンです。サブCPUがプリントをするときにエラーが発生した場合、C_{82}-BASICのエラー処理を実行します。

②XCOPY(0A4CH)

(機能)　画面上の巾8bit縦一列のビットイメージをプリンタへ出力します。

(IN条件)　Areg　色コード(bitが1の画面のみ扱います)

Creg　X座標(文字座標で指定します)

(OUT条件)　なし

(説明)　スクリーンコピー機能です。転送は画面の下から上へ行われます。また、プリンタのコントロールシーケンスは、このサブルーチンに先立って行わなければなりません。

8-4　CMT(カセットテープレコーダ)

①XMTON(0A73H)

(機能)　CMTのリモートスイッチをONにします。

(IN条件)　なし

(OUT条件)　なし

②XMTOFF(0A78H)

(機能)　CMTのリモートスイッチをOFFにします。READ／WRITE動作の終了をサブCPUに通知します。CMT動作中に禁止していた割込処理を許可します。

(IN条件)　なし

(OUT条件)　なし

③XMTMD(0A91H)

(機能)　CMTのボーレートを指定します。

(IN条件)　Areg　{ 00H　1200ボー
　　　　　　　　　 01H　300ボー

(OUT条件)　なし

④XMTSY(0A9AH)

(機能)　CMTにREAD動作の開始を指示します。

(IN条件)　なし

(OUT条件)　なし

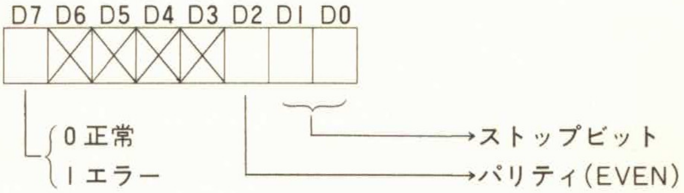
⑤XMTRD(0AA2H)

(機能)　CMTから1byte読取ります。

(IN条件)　なし

(OUT条件)　Breg　データ(8bit)

Creg



⑥XMTWR(0AB5H)

(機能)　CMTに1byte出力します。

(IN条件)　Areg　出力データ

(OUT条件)　なし

⑦XMTFF(0AE0H)

（機能）　　　12bitのALL1を出力します。IRGの作成に使用します。

（IN条件）　　なし

（OUT条件）　なし

※CMT動作を開始すると、全ての他のデバイスの動作が禁止されます。また、XMTOFルーチンによってCMT動作が終了するまでは、CMT以外の機能を起動することは禁じられます。

8-5 RS-232C

①XSRES(0605H)

(機能)	RS-232Cパック内のUSARTをイニシャライズします。	
(IN条件)	Breg	コマンドコード
	Creg	モードコード
	Dreg	スロット番号(0~7)
(OUT条件)	なし	
(説明)	該当スロットにRS-232Cパックが無い場合、または、300ms経過してDSRがONにならない場合、C_{82}-BASIC内のエラー処理を実行します。	

②XRSOF(0629H)

(機能)	送受信を禁止し、DTRをOFFにします。送信が終了しない場合は、300msの時間待ちの後、打ち切ります。	
(IN条件)	Dreg	スロット番号(0~7)
(OUT条件)	なし	

③XRSBK(0636H)

(機能)	300msecのBREAK信号を出力します。
(IN条件)	なし
(OUT条件)	なし

④SOUT1(064EH)

(機能)	データを出力します。		
(IN条件)	Areg	出力データ	
	Dreg	スロット番号(0~7)	
(OUT条件)	Cflag	0	OK
		1	300mscc経過してもデータ転送が終了しない。

⑤SIN

RS-232Cの入力は割込処理のため、BASICでの"OPEN"、"CLOSE"等と密接に結びついており特定のサブルーチンとしての機能は存在しません。

以下に割込を使用しない、1byte入力ルーチンの例を示します。

なお、このルーチンを使用する際には、前述のポートセットルーチンを使って、RS-

232Cパックを選択しておいて下さい。

また、通常の状態ではOSが、RS-232Cを、割込処理としてサポートしますので、競合を避けるためマシン語処理の先頭に

```
CALL   0AE6H   (RS-232Cの割込を禁止)
LD     BC,0FF80H
OUT    (C),A
```

また処理の最後に［CALL　0A7DH］(RS-232C割込の許可)を必ず実行して下さい。

```
SIN1:   CALL    SINC
        JR      NC,SIN1
        RET
SINC:   PUSH    BC
        LD      BC,0001H
        IN      A,(C)
        SRL     A
        SRL     A
        JR      NC,S301
        DEC     C
        IN      A,(C)
        INC     C
S301:   PUSH    AF
        LD      A,37H
        OUT     (C),A
        POP     AF
        POP     BC
        RET
```

8-6 FDD

①RECRB(032EH)

(機能)	ヘッドを0トラックへ戻します。	
(IN条件)	Dreg	スロット番号(0~7)
	Ereg	ドライブ番号(0または1)
(OUT条件)	なし	

②FMT(033AH)

(機能)	イニシャライズを行います。FORMAT文と同じです。	
(IN条件)	Dreg	スロット番号(0~7)
	Ereg	ドライブ番号(0または1)

③SECRD(0356H)

(機能)	1セクタ読出します。	
(IN条件)	Breg	セクタ 番号(0~31, 16以上はサイド1)
	Creg	トラック(0~39)
	Dreg	スロット番号(0~7)
	Ereg	ドライブ番号(0または1)
	HLreg	データ格納エリア(256byte)の先頭
(OUT条件)	なし	

④SECWR(0372H)

(機能)	1セクタ書込みます。	
(IN条件)	Areg	00H 書込みのみ 他 リードアフタライト指定
	Breg	セクタ 番号(0~31, 16以上はサイド1)
	Creg	トラック番号(0~39)
	Dreg	スロット番号(0~7)
	Ereg	ドライブ番号(0または1)
	HLreg	書込むデータの先頭番地

※FDD動作中にエラーが発生した場合、C_{82}-BASICのエラー処理が行なわれます。

8-7　演算サブルーチン

演算サブルーチンは主に、 FA1(9E7FH～9E91H),FA2(9E92H～9EA4H) に格納されたデータを扱います。また、場合によっては、各々の一部である。

FA1S(9E81H～9E91H),FA2S(9E94H～9EA4H)　のデータも扱います。

数値の内部表現は、「5-1 BASIC内のデータ表現」で紹介した表現型式の先頭に精度情報を1byte付けた、E形式と、主に関数演算などで扱うF形式があります。F形式についての説明は割愛いたしますが、以降の節でE型式とF型式の相互変換サブルーチンを紹介します。

また、E形式の先頭にある精度情報は以下の通りです。

02H	整数型
06H	単精度実数型
0BH	倍精度実数型
10H	倍々精度実数型

8-7-1　型変換

①CNV(3BD3H)

(機能)　任意の精度から任意の精度へ変換します。

(IN条件)　Areg　変換したい型の精度情報
　HLreg　E形式のデータの先頭アドレス

(OUT条件)　HLregによって指定されたアドレス以降にE形式のデータ

②CNVI(5959H)

(機能)　整数型に変換します。

(IN条件)　FA1S　データ(E形式)

(OUT条件)　FA1S　データ(E形式)

③CNVS(594DH)

(機能)　単精度実数型に変換します。

(IN条件)　FA1S　データ(E形式)

(OUT条件)　FA1S　データ(E形式)

④CNVD(5951H)

(機能)　倍数度実数型に変換します。

(IN条件)　FA1S　データ(E形式)

（OUT条件）　FA1S　データ（E形式）

⑤CNVF（5955H）

（機能）　倍々精度実数型に変換します。

（IN条件）　FA1S　データ（E形式）

（OUT条件）　FA1S　データ（E形式）

⑥CNVEF1（468EH）

（機能）　E形式をF形式へ変換します。

（IN条件）　FA1S　データ（E形式）　但し整数型は除く。

（OUT条件）　FA1S　データ（F形式）

⑦CNVEF2（4689H）

（機能）　E形式をF形式へ変換します。

（IN条件）　FA2S　データ（E形式）　但し整数型は除く。

（OUT条件）　FA2S　データ（F形式）

⑧CNVFE（46B1H）

（機能）　F形式をE形式へ変換します。

（IN条件）　FA1　データ（F形式）

（OUT条件）　FA1　データ（E形式）

8-7-2　四則演算

四則演算は、エントリーポイントの違いのみで、IN条件、OUT条件は同じです。また、計算の結果は丸め処理をされています。

（IN条件）　FA1S　被演算数（E形式）

FA2S　演算数（E形式）

（OUT条件）　FA1S　演算結果（E形式）

①ADD（47F2H）　加算

②SUB（480DH）　減算

③MUL（4815H）　乗算

④DIV（482AH）　除算

⑤PWR（470DFH）　べき乗算

8-7-3　関数

関数もエントリーポイントの違いのみで、IN条件、OUT条件は同じです。また計算の結果は丸め処理されていません。必要な場合は、ROUNDサブルーチンで結果を丸めて下さい。

（IN条件）　FA1　引数（F形式）

（OUT条件）　FA1　結果（F形式）

エントリーポイントは下表の通りです。

関　数	エントリーポイント	関　数	エントリーポイント
SQR	4A00H	HCN	5042H
SIN	55D2H	HTN	5048H
COS	55CEH	AHS	503FH
TAN	55CAH	AHC	5045H
ASN	5677H	AHT	504BH
ACN	5673H	LOG	5091H
ATN	566FH	LGT	5495H
HSN	503CH	EXP	5526H

①ROUND（4E09H）

（機能）　丸め処理をします。

（IN条件）　FA1　データ（F形式）

（OUT条件）　FA1　データ（F形式）

8-7-4　文字数値変換

①CNVCN（593AH）

（機能）　文字で表現されたデータを内部形式に変換します。C_{82}-BASICではVAL関数に相等します。

（IN条件）　DEreg　データの先頭アドレス。データの最後は0DHが必要です。

（OUT条件）　Cflag　{ 0　変換不能 / 1　OK }

FAIS　データ（E形式）

②CNVNC（6004H）

（機能）　内部形式で表現されたデータを文字での表示形式に変換します。

（IN条件）　FAIS　データ（E形式）

（OUT条件）　Breg　文字数

HLreg　変換されたデータの先頭アドレス

CASIO

カシオ計算機株式会社 営業本部

〒160 東京都新宿区西新宿2-6-1 新宿住友ビル

TEL 03—347—4811（代表）

昭和57年10月1日現在　M017-10282B